

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) An information processing apparatus for processing data, comprising a plurality of operating systems and a processor and memory, the plurality of operating systems including a main operating system controlling an interrupt process and a sub operating system,

the main operating system, along with a system control operating system set as the sub operating system:

setting a logical partition as a process unit,

enforcing a customizable upper limit on the resources available to the logical partition that one or more guest sub operating systems can allocate or release without communicating with the system control operating system, and

managing a hardware resource relating to the logical partition;

the sub operating system:

operating within the logical partition set by the main operating system and the system control operating system, and

executing a software application program with the hardware resource assigned to the logical partition; and

the main operating system:

storing, in the memory, status information as to whether the sub operating system is in an interrupt-enabled state or an interrupt-disabled state, and

controlling the interrupt process, based on the status information, to perform one of an interrupt process execution and an interrupt process reserve, in response to generation of the interrupt.

2. (Previously Presented) The information processing apparatus according to claim 1, wherein the main operating system stores interrupt process status information as to whether the interrupt process is in progress or in reserve, and resumes the interrupt process execution in response to the transition of the sub operating system between the interrupt-enabled state and the interrupt-disabled state.

3. (Previously Presented) The information processing apparatus according to claim 1, wherein the sub operating system notifies the main operating system of the status information as to whether the sub operating system is in the interrupt-enabled state or the interrupt-disabled state, and

wherein the main operating system updates the status information of the sub operating system in response to the notification from the sub operating system.

4. (Original) The information processing apparatus according to claim 1, wherein the main operating system stores priority information of the interrupt process, and performs the interrupt process responsive to the priority information.

5. (Previously Presented) The information processing apparatus according to claim 1, wherein the main operating system performs status management based on a status table containing the status information of the sub operating system and the interrupt process status information as to whether the interrupt process is in progress or in reserve,

if an interrupt intended for the sub operating system is generated and the main operating system determines based on the status table that the sub operating system is in the interrupt-disabled state, registers the interrupt in the status table as a reserved interrupt, and

if an interrupt intended for the sub operating system is generated and the main operating system determines based on the status table that the sub operating system is in the interrupt-enabled state, performs interrupt control depending on whether the operating system operating on a processor is either the main operating system or the sub operating system in a manner such that:

(a) if the main operating system is in operation, the main operating system

(a1) executes the interrupt process in response to a high-priority interrupt,

or

(a2) reserves the interrupt process in response to a low-priority interrupt;

and that

(b) if the sub operating system is in operation, the main operating system executes the interrupt process regardless of the priority level of the interrupt.

6. (Previously Presented) The information processing apparatus according to claim 1, wherein the main operating system performs status management based on a status table containing the status information of the sub operating system and the interrupt process status information as to whether the interrupt process is in progress or in reserve, and

if an interrupt intended for the main operating system is generated, performs interrupt control depending on whether the operating system operating on a processor is either the main operating system or the sub operating system in a manner such that

(a) if the main operating system is in operation, the main operating system executes the interrupt process regardless of the priority level of the interrupt, and that

(b) if the sub operating system is in operation, the main operating system

(b1) executes the interrupt process in response to a high-priority interrupt, or

(b2) reserves the interrupt process in response to a low-priority interrupt.

7. (Currently Amended) An interrupt process control method for performing data processing on a plurality of operating systems, the method comprising steps of:

in a main operating system: [[,]]

controlling an interrupt process,

managing a hardware resource to a logical partition, with the logical partition being handled as a process execution unit in the main operating system along with a system control operating system set as a sub operating system, and

enforcing a customizable upper limit on the resources available to the logical partition that one or more guest sub operating systems can allocate or release without communicating with the system control operating system;

in the sub operating system, executing a software application program, with the hardware resource assigned to the logical partition being used, within the logical partition set by the main operating system and the system control operating system;

in the main operating system, receiving, from a sub operating system other than a main operating system, status information as to whether the sub operating system is in an interrupt-enabled state or an interrupt-disabled state;

detecting the generation of an interrupt; and

controlling an interrupt process, based on the status information, to perform one of an interrupt process execution and an interrupt process reserve, in response to the generation of the interrupt.

8. (Previously Presented) The interrupt process control method according to claim 7, wherein the main operating system stores interrupt process status information as to whether the interrupt process is in progress or in reserve, and resumes the

interrupt process execution in response to the transition of the sub operating system between the interrupt-enabled state and the interrupt-disabled state.

9. (Previously Presented) The interrupt process control method according to claim 7, further comprising steps of:

notifying the main operating system of the status information as to whether the sub operating system is in the interrupt-enabled state or the interrupt-disabled state, and

updating the status information of the sub operating system in response to the notification from the sub operating system.

10. (Original) The interrupt process control method according to claim 7, wherein the main operating system stores priority information of the interrupt process, and performs the interrupt process responsive to the priority information.

11. (Previously Presented) The interrupt process control method according to claim 7, wherein the main operating system performs status management based on a status table containing the status information of the sub operating system and the interrupt process status information as to whether the interrupt process is in progress or in reserve,

if an interrupt intended for the sub operating system is generated and the main operating system determines based on the status table that the sub operating system is in the interrupt-disabled state, registers the interrupt in the status table as a reserved interrupt, and

if an interrupt intended for the sub operating system is generated and the main operating system determines based on the status table that the sub operating system is

in the interrupt-enabled state, performs interrupt control depending on whether the operating system operating on a processor is either the main operating system or the sub operating system in a manner such that:

(a) if the main operating system is in operation, the main operating system

(a1) executes the interrupt process in response to a high-priority interrupt,

or

(a2) reserves the interrupt process in response to a low-priority interrupt;

and that

(b) if the sub operating system is in operation, the main operating system executes the interrupt process regardless of the priority level of the interrupt.

12. (Previously Presented) The interrupt process control method according to claim 7, wherein the main operating system performs status management based on a status table containing the status information of the sub operating system and the interrupt process status information as to whether the interrupt process is in progress or in reserve, and if an interrupt intended for the main operating system is generated, performs interrupt control depending on whether the operating system operating on a processor is either the main operating system or the sub operating system in a manner such that

(a) if the main operating system is in operation, the main operating system executes the interrupt process regardless of the priority level of the interrupt, and that

(b) if the sub operating system is in operation, the main operating system

(b1) executes the interrupt process in response to a high-priority interrupt, or

(b2) reserves the interrupt process in response to a low-priority interrupt.

13. (Currently Amended) A computer program, embodied in a non-transitory computer-readable medium, for performing data processing on a plurality of operating systems, the computer program, when executed on one or more processors, causing the one or more processors to perform steps comprising:

in a main operating system;

controlling an interrupt process,

managing a hardware resource to a logical partition, with the logical partition being handled as a process execution unit in the main operating system along with a system control operating system set as a sub operating system, and

enforcing a customizable upper limit on the resources available to the logical partition that one or more guest sub operating systems can allocate or release without communicating with the system control operating system;

in the sub operating system, executing a software application program, with the hardware resource assigned to the logical partition being used, within the logical partition set by the main operating system and the system control operating system;

in the main operating system, receiving, from a sub operating system other than an main operating system, status information as to whether the sub operating system is in an interrupt-enabled state or an interrupt-disabled state;

detecting the generation of an interrupt; and

controlling an interrupt process, based on the status information, to perform one of an interrupt process execution and an interrupt process reserve, in response to the generation of the interrupt.